Gesture-based Numeral Extraction and Recognition

Shree Prakash

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

Gesture-based Numeral Extraction and Recognition

Thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Technology

(Research)

in

Computer Science and Engineering

by

Shree Prakash

(Roll: 611CS106)

under the guidance of

Prof. Banshidhar Majhi

&

Dr. Pankaj Kumar Sa

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela-769 008, Odisha, India

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

December 22, 2014

Certificate

This is to certify that the work in the thesis entitled ***Gesture-based Numeral Extraction and Recognition*** by ***Shree Prakash*** (roll number 611CS106), is a record of an original research work carried out under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Master of Technology (Research) in Computer Science and Engineering. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Pankaj Kumar Sa
Assistant Professor

Banshidhar Majhi
Professor

Acknowledgment

I owe deep gratitude to the ones who have contributed greatly in completion of this thesis.

Foremost, I would like to express my sincere gratitude to my advisor, Prof. Banshidhar Majhi for providing motivation, enthusiasm, and critical atmosphere at the workplace. His profound insights and attention to details have been true inspirations to my research.

I would like to thank Prof. Pankaj Kumar sa for his constructive criticism during entire span of research. His insightful discussions has helped me a lot in improving this work.

I am very much indebted to Prof. Kishore Chandra Pati, Prof. Pabitra Mohan Khilar, Prof. Susmita Das, and Prof. Gopal Krishna Panda for providing insightful comments at different stages of thesis that were indeed thought provoking.

I would like to thank all my friends and lab-mates for their encouragement and understanding. Their help can never be penned with words.

Most importantly, none of this would have been possible without the love and patience of my family. My family to whom this dissertation is dedicated to, has been a constant source of love, concern, support and strength all these years. I would like to express my heart-felt gratitude to them.

Shree Prakash

Abstract

In this work the extraction of numerals and recognition is done using gesture. Gestures are elementary movements of a human body part, and are the atomic components describing the meaningful motion of a person. It is of utmost importance in designing an intelligent and efficient human-computer interface. Two approaches are proposed for the extraction of numeral from gesture. In the first approach, numerals are formed using the finger gesture. The movement of the finger gesture is identified using optical flow method. A view-specific representation of movement is constructed, where movement is defined as motion over time. A temporal encoding is performed from different frames into a single frame. To achieve this we utilize motion history image (MHI) scheme which spans the time scale of gesture. In the second approach, gesture is performed by the use of a pointer like a pen whose tip is either red, green, or blue. In the scene multiple persons are present performing various activities, but our scheme only captures the gesture made by the desired object. HSI color model is used to segment the tip followed by the optical flow to segment the motion. After getting the temporal template, the features are extracted and the recognition is performed. Our second approach is invariant to uninteresting movements in the surrounding while capturing the gesture. Hence it will not affect the final result of recognition.

Keywords: Numeral recognition, Gesture, Optical flow, Motion history image, HSI color model.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Human activity recognition is an important area of computer vision research. There are various types of human activities, which can be divided into four different levels: gestures, actions, interactions, and group activities [1]. Gestures are elementary movements of a human body part, and are the atomic components describing the meaningful motion of a person. Motion of finger, stretching an arm, and raising a leg are some examples of gestures. Actions are single-person activities that may be composed of multiple gestures organized temporally, such as walking, waving, and punching. Interactions are human activities that involve two or more persons and/or objects, for example, fighting between two persons is an interaction, whereas, a person, stealing a suitcase from another person is a human-object interaction. Group activities represent activities performed by groups, composed of multiple persons and/or objects, for instance a discussion of an event among the committee members.

Gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or body with the intent of conveying meaningful information or interacting with the environment. They constitute one interesting small subspace of possible human motion. A gesture may also be perceived by the environment as a compression technique for the information to be transmitted elsewhere and subsequently reconstructed by the receiver. Gesture recognition has wide-ranging applications such as developing aids for the hearing impaired, recognizing sign language, lie detection, designing techniques for forensic identification. However gestures are ambiguous and incompletely specified. For example, to indicate the concept stop, one can use gestures such as a raised hand with

palm facing forward or a waving of both hands over the head. Gestures can be static or dynamic. In static based gesture, the user assumes a certain configuration, whereas the dynamic ones are associated with pre-stroke, stroke and post-stroke phases. Some gestures also have both static and dynamic elements, as in sign languages. Gestures are often language and culture specific. Broad classification is listed in Table 1. The meaning of a gesture can be dependent on (a) spatial information: where it occurs (b) temporal information: the path it takes (c) symbolic information: the sign it makes (d) affective information: its emotional quality. The same gesture may dynamically vary in shape and duration even for the same person.

Table 1.1: Classification of gesture

Classification	Gesture details
hand and arm gesture	recognition of hand poses, sign languages
head and face gesture	nodding or shaking of head, direction of eye gaze, raising the eyebrows, opening the mouth to speak, winking, flaring the nostrils, looks of surprise, happiness, disgust, fear, anger, sadness
body gesture	involvement of full body gesture as tracking movement of two people interacting outdoors, analyzing movements of a dancer for generating matching music and graphics

Gesture recognition is an ideal example of multidisciplinary research. Human gestures typically constitute a space of motion expressed by the body, face, and/or hands. Gesture may be categorized as given in the following list:

- *Gesticulation*: spontaneous movement of hands and arms, accompanying speech. These spontaneous movements constitute around 90% of human gestures. People gesticulate when they are on telephone, and even blind people regularly make gesture when speaking to one another.

- *Languagelike gestures*: gesticulation integrated to a spoken utterance, replacing a particular spoken word or phrase.

- *Pantomimes*: gestures depicting objects or actions, with or without accompanying speech.

- *Emblems*: familiar signs such as V for victory or other culture-specific gestures.

- *Sign languages*: well-defined linguistic systems. These carry the most semantic information and are more systematic, thereby easier to model in a virtual environment.

In this thesis, our focus is to identify numerals of any language through a finger tip gesture both marked with color or without color.

1.1 Related Work

Bobick *et al.* [2] proposed a view based approach, in which a temporal template, describes where the motion is and the pattern of movement. View-specific representation of movement is constructed, where movement is defined as temporal motion over frames, assuming that either the background is static, or the motion of an object can be separated from the camera-induced or distractor motion. Aggarwal *et al.* [1] have summarized the different methodologies for the recognition of human activity, and discussed the advantages and disadvantages of different approaches. All activity recognition methodologies are classified into two categories (a) single-layered approaches and (b) hierarchical approaches. Single-layered approaches recognize human activities directly based on sequence of images. Hierarchical approaches represent high-level human activities in terms of other simpler activities, which are called sub-events. Mitra *et al.* [3] have described briefly the different tools and their use for gesture recognition. Shan *et al.* [4] have proposed a novel approach for hand gesture recognition. The spatio-temporal trajectory of hand gesture is tracked, and then represented in a static image using temporal template. Ishikawa *et al.* [5] have done the recognition of hand gesture using a data glove which measures the angle between the finger joint. It has two sensors for the first and the second joint of each finger, in total it has ten angle sensors. A resulting ten dimensional vector represents a hand shape. Qureshi *et al.* [6] have proposed an algorithm for human hand gesture identification. The core work is the identification of finger which are active and those which are not. The peak or apex type pattern in fingers are identified which are regarded as joints of fingers. This joint detection is used to identify the active fingers. Sohn *et al.* [7] have proposed a 3D hand gesture recognition scheme, in

which the hand gesture video is obtained from 3D depth camera. Ahad *et al.* [8] have presented a temporal motion segmentation method, based on directional motion history templates in which optical flow is calculated and sectioned into four channels based on four directions: up, down, left, and right, and recognition of different actions such as body stretching, waving arms, bending the chest has been performed. The optical flow [9, 10] of a pixel is a motion vector represented by the motion between a pixel in one frame and its correspondence pixel in the following frame. In [11] each gesture is defined to be an ordered sequence of state in spatio-temporal space. The 2D image positions of the center of the head and both hands are used as features; these are located by a color based tracking method. Lei *et al.* [12] have devised an accelerometer-based method for detecting the predefined one-stroke finger gestures, where data is collected using a MEMS 3D accelerometer, worn on the index finger. A compact wireless sensing mote integrated with the accelerometer, called magic ring, is developed to be worn on the finger for real data collection. A general definition on one-stroke gesture is given, and twelve kinds of one-stroke finger gestures are selected from human daily activities. Cemil *et al.* [13] have developed an American Sign Language(ASL) recognition system. It uses a sensory glove called the cyber glove and a flock of birds to extract the gesture feature. The glove has eighteen sensors, which measure the bending angles of fingers at various positions. There are fifteen sensors on the glove: three sensors for the thumb, two sensors for each of the other four fingers, and four sensors between the fingers. To track the position and orientation of the hand in 3D space, the flock of birds motion tracker is mounted on the hand and wrist is used.

1.2 Motivation

It is observed that gesture is the natural form of communication. Controlling the home appliance, interaction with the computer is achieved using gesture. Gesture works even in the presence of sound noise. Even dumb people can use gesture to communicate with device. Usually hands and fingers are used to make numeral gesture. External device like data glove requires the user to wear a cumbersome

device and carry a load of cables connecting the device to a computer. From the literature, it is noted that sensors are used to make gestures and optical flow is a popular method to identify the motion of objects.

1.3 Objectives

In this thesis we have investigated to identify numerals through a gesture made by a fingertip or using pointer having a colored tip. In particular, the objectives of suggested scheme are narrowed to:

(a) Capture the gesture from a dynamic environment.

(b) Motion segmentation using optical flow mechanism.

(c) Formation of temporal template of numeral.

(d) Feature extraction.

(e) Classification and recognition of the extracted numeral.

The overall block structure is given in Figure 1.1 and phases are discussed below in nutshell.

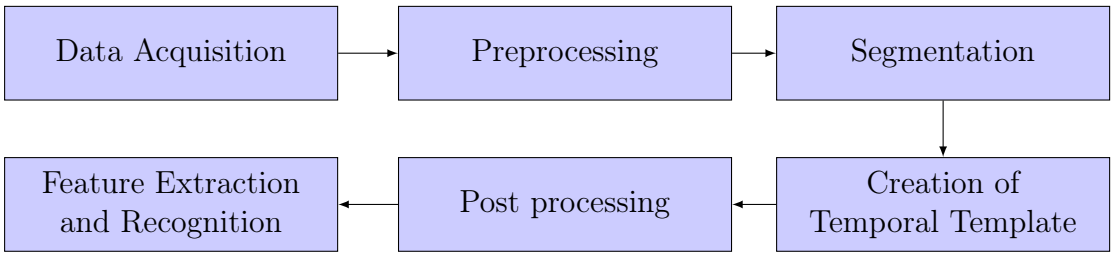


Figure 1.1: Block diagram for the recognition of numeral using gesture.

(a) **Data acquisition :**

The presence of the five sensory organs of a human body helps to interact, learn and adapt with the challenging environment. The sight sensory organ helps in receiving visual information. This visual information can be captured and stored as an image by a camera. A single image is inadequate enough

to represent a scene with motion information. Such scenes are recorded by capturing a sequence of images at regular intervals. Each image of the sequence is known as frame. When successive frames are projected with the progress of time, we call it as *video*. Projection of successive frames at a particular rate creates an illusion, which convey a sense of motion in the scene.

(b) **Segmentation :**

In computer vision, segmentation refers to the process of partitioning a digital image into multiple segments. It is the allocation of every pixel in an image, a label to which they correspond to a specific part. The goal of image segmentation is to partition the image into perceptually similar regions [14–16]. Segmentation is an extremely important operation in several applications of image processing and computer vision, since it represents the very first step of low-level processing of imagery [10, 17]. Every segmentation algorithm addresses two problems, the criteria for a good partition and the method for achieving efficient partitioning [18,19]. All image processing operations generally aim at a better recognition of objects of interest, i.e., finding suitable local features that can be distinguished from other objects and from the background. The next step is to check each individual pixel, whether it belongs to an object of interest or not. Image segmentation produces a binary image, where one represents the object and zero represents the static background. There are three general approaches to segmentation: thresholding, edge-based methods and region-based methods [20]. In thresholding, pixels are allocated to categories according to the range of values in which a pixel lies. In edge-based segmentation, an edge filter is applied to the image, and pixels are classified as edge or non-edge depending on the filter output, and pixels which are not separated by an edge are allocated to the same category. Region-based segmentation algorithms operate iteratively by grouping pixels which are neighbors and have similar values and splitting groups of pixels which are dissimilar in value.

In this thesis, segmentation of motion and color is carried out using thresholding mechanism. Motion is an integral part of video sequence. It is an essential

building block for robotics, inspection, metrology, visual surveillance, video indexing and many other applications. It provides a very rich set of information through which a wide variety of works are accomplished. Perceptual organization, 3D shape determination, scene understanding are to name a few. Motion-based segmentation algorithms generally involves three main issues. The first issue is data primitives or region of support [21], the data primitives can be individual pixels, corners, lines, blocks or regions. The second issue is motion models or motion representations, which can be 2D optical flow, or 3D motion parameters, which involves parameter estimation or motion estimation. The third issue is segmentation criteria. The main attributes of a motion segmentation algorithm can be summarized as follows.

- $Feature-based$ or $Dense-based$: In Feature-based methods, the objects are represented by a limited number of points like corners or salient points, whereas Dense-based methods compute a pixel-wise motion.
- $Multiple\ objects$: ability to deal with more than one object in the scene.
- $Spatial\ continuity$: ability to exploit spatial continuity.
- $Temporary\ stopping$: ability to deal with temporary stop of the objects.
- $Robustness$: it is the ability to deal with noisy images (in case of feature based methods it is the position of the point to be affected by noise but not the data association).

In this thesis data primitives is individual pixel and motion is represented using 2D optical flow.

Color is one of the most distinctive clues in finding objects. Several color representations are currently used in color image processing. The most common is the RGB space where colors are represented by their red, green, and blue components in an orthogonal Cartesian space. This is in agreement with the $tristimulus\ theory\ of\ color$ [22,23] according to which the human visual system acquires color imagery by means of three band pass filters (three different kinds

of photoreceptors in the retina called cones) whose spectral responses are tuned to the wavelengths of red, green, and blue.

(c) **Temporal Template :**

Appearance-based motion recognition methods is one of the most practical recognition methods for identifying a gesture without any incorporation of sensors on the human body or its neighborhood. A view-specific representation of movement is constructed, where movement is defined as motion over time. The image sequence is converted into a static shape pattern [8].

(d) **Post processing :**

Some morphology operations [20] such as Dilation, Erosion, Thinning, Pruning are employed in order to have a invariant and stable representation.

- Dilation: an operation that grows or thickens object in an image.
- Erosion: an operation that shrinks or thins object in a binary image.
- Thinning: an operation in which binary valued image regions are reduced to lines that approximate the center skeletons of the regions [24] . It gives the skeleton representation of object that preserves the topology aiding synthesis and understanding.
- Pruning: an operation in which spur outliers are removed by setting pixel values to black. It is implemented by detecting end points and by removing them until idempotence [25].

(e) **Feature Extraction and Recognition :**

The feature is defined as a function of one or more measurements, each of which specifies some quantifiable property of an object, and is computed such that it possesses some significant characteristics of the object [26]. The classification of various features is given as follows:

- **General features**: Application independent features such as color, texture, and shape. According to the abstraction level, they can be further divided into:

- **Pixel-level features**: Features calculated at each pixel, e.g. color, location.
- **Local features**: Features calculated over the results of subdivision of the image band on image segmentation or edge detection.
- **Global features**: Features calculated over the entire image or just regular sub-area of an image.

- **Domain-specific features**: Application dependent features such as human faces, fingerprints, and conceptual features. These features are often a synthesis of low-level features for a specific domain.

On the other hand, all features can be coarsely classified into low-level features and high-level features. Low-level features can be extracted direct from the original images, where as high-level feature extraction must be based on low level features [27].

1.4 Thesis Organization

The overall thesis is organized into five chapters including the introduction.

Chapter 2 presents the formation of numeral using finger gesture. Motion of finger is obtained using 2D motion vector. Temporal template of the numeral are formed and morphological operation are performed. Motion of finger is captured through a video, but the other moving parts are removed to extract the motion of the finger.

Chapter 3 presents the formation of numeral using a pen whose tip is either red or green or blue. In the scene multiple persons are present performing different activity. however the extraction of the desired tip is extracted.

Chapter 4 deals with the feature extraction from the final template and its recognition performance is studied.

Chapter 5 presents the concluding remarks with scope for future research work.

Chapter 2

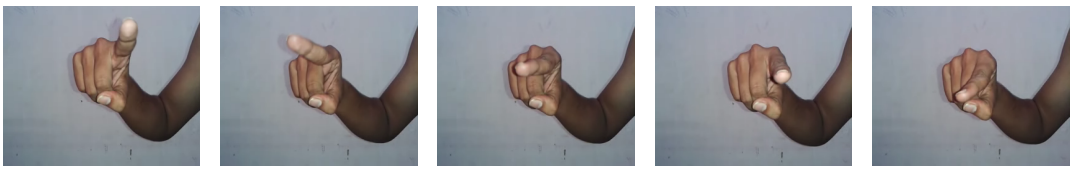
Formation of Numeral using Finger Gesture

In this chapter, we exploit all the phases/steps required for numeral formation using finger gesture. The steps in order are given below.

- Video data acquisition
- Motion segmentation using optical flow method
- Motion history image formation

2.1 Video Acquisition

Video are taken as input data in our system. In the video the motion of index finger is captured using a mobile camera having resolution 5M pixel. The motion is in such a way that it makes the gesture of a particular numeral as shown in Figure 2.1, which symbolize numeral '2' in terms of frames.



(a) Frame 1 (b) Frame 20 (c) Frame 30 (d) Frame 60 (e) Frame 84

Figure 2.1: Frames of finger gesture of original video.

Camera stores the video in inverted form. The input video is preprocessed to

convert it into a normal form prior to motion segmentation using Algorithm 1 and the result is shown in Figure 2.2.

(a) Frame 1 (b) Frame 20 (c) Frame 30 (d) Frame 60 (e) Frame 84

Figure 2.2: Frames of finger gesture of preprocessed video.

Algorithm 1 Rectification of original video

Input : Original Video O
Output : Vertical mirror imaged video U
$w \leftarrow number\ of\ column\ of\ each\ frame$
$c \leftarrow w$
for $k \leftarrow 1$ *to* $number\ of\ frame$ **do**
  for $i \leftarrow 1$ *to* $height\ of\ frame$ **do**
    for $j \leftarrow 1$ *to* $width\ of\ frame$ **do**
      $U(i, j, :, k) \leftarrow O(i, w, :, k)$
      $w \leftarrow w - 1$
    end for
    $w \leftarrow c$
  end for
end for

2.2 Motion Segmentation

Motion segmentation aims at decomposing a video in moving objects and background [28]. When an object, moving along a path in a three-dimensional co-ordinate system, is projected on an image plane, each point produces a two-dimensional path. Its instantaneous direction is its velocity and the 2D velocities at all such points is usually known as 2D motion field. Methods used in moving object detection are mainly the frame subtraction method, the background subtraction method and the optical flow method. The background subtraction approach is to use the difference method of the current frame and the background frame to detect moving objects. In the frame subtraction method the presence of moving objects is determined by calculating the

difference between two consecutive frames. Obviously, these two methods cannot be applied to this type of particular gestures. So in this case, optical flow mechanism is applied

Optical flow method is employed to estimate an approximation of the motion field from a set of images varying with respect to time. It is a 2D vector which gives the displacement of each pixel with respect to its previous frame. In other words optical flow is the distribution of apparent velocities of movement of brightness pattern [9]. It arises due to the relative motion of object and viewer. If the camera, or an object, moves within the scene, this motion results in a time dependent displacement of the gray values in the image sequence. The resulting two-dimensional apparent motion field in the image domain is called the optical flow field. There are various methods to compute optical flow given in literature [29–35]. It is a dense field of displacement vectors which defines the translation of each pixel in a region [19].

2.2.1 Computation of optical flow

Three popular methods to compute optical flow are Horn and Schunck method [9], Lucas and Kanade Window method (LKW) [36], and Least Square Fit Method (LKF) [37] are implemented towards the simulation of the proposed work discussed in detail below in sequence.

(a) **Horn and Schunck method**: In this method computation of optical flow is based on two assumptions: brightness constancy and velocity smoothness. For better understanding we describe both below in nutshell.

- **Brightness constancy**: The observed brightness of any object point is constant over time. Let F(x, y, t) is brightness at image point (x, y) at time t and image moves dx in x-direction, dy in y-direction during interval dt, then

$$F(x + dx, y + dy, t + dt) = F(x, y, t) \tag{2.1}$$

  Using Taylor series expansion and neglecting higher order terms yields

$$(\partial F/\partial x).dx + (\partial F/\partial y).dy + (\partial F/\partial t).dt = 0 \tag{2.2}$$

For simple notation, let

$$(\partial F/\partial x) = f_x,\ (\partial F/\partial y) = f_y,\ (\partial F/\partial t) = f_t$$

Using this notation and dividing equation (2.2) with dt, we get

$$f_x.dx/dt + f_y.dy/dt + f_t = 0 \tag{2.3}$$

Let

$$dx/dt = u,\ dy/dt = v$$

So the equation (2.2) became

$$f_x.u + f_y.v + f_t = 0 \tag{2.4}$$

where u and v is the velocity components of each pixel in the x and y direction and $(f_x,\ f_y)$ is the rate of change of brightness with respect to time at a point in the image. Figure 2.3 shows equation (2.4) is a straight line with u as x-axis

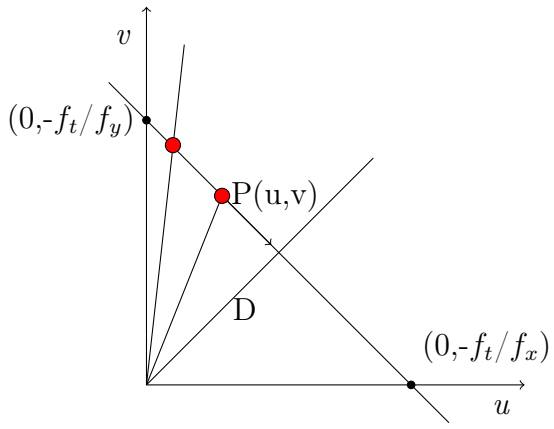


Figure 2.3: Interpretation of optical flow equation.

and v as y-axis. Optical flow of point P can be anywhere on the straight line. Point P has two types of flow; parallel flow, which is along the straight line and normal flow, perpendicular to the straight line. Normal flow is not changing, the distance D remains constant, however parallel flow is changing which need to be computed. In equation (2.4), f_x, f_y, and f_t are known and unknown variables are u and v, so it is an under constrained equation. To get the unknown variables

u and v at least two equations are required, i.e. an additional constraint is required.

- **Velocity smoothness**: Nearby points in the image plane move in a similar manner. One way to express this constraint is to minimize the square of the magnitude of the gradient of the optical flow velocity.

$$(\partial u/\partial x)^2 + (\partial u/\partial y)^2 \; and \; (\partial v/\partial x)^2 + (\partial v/\partial y)^2 \tag{2.5}$$

Ideally equation (2.4) has to be zero, but in practical scenario it is not, also there is deviation from smoothness in the velocity flow given in equation (2.5), so the total error to be minimized is:

$$\int\int (f_x.u + f_y.v + f_t)^2 + ((\partial u/\partial x)^2 + (\partial u/\partial y)^2 + (\partial v/\partial x)^2 + (\partial v/\partial y)^2)dxdy \tag{2.6}$$

solving the equation (2.6), we get

$$(f_x.u + f_y.v + f_t).f_x + \alpha(u - u_{avg}) = 0 \tag{2.7}$$

$$(f_x.u + f_y.v + f_t).f_y + \alpha(v - v_{avg}) = 0 \tag{2.8}$$

where α is a smoothness constraint. In order to compute u_{avg} and v_{avg} for each pixel find its 4-neighborhood, add all the pixel value and divide the sum by 4.

(b) **Lucas and Kanade Window method (LKW)**: Equation (2.4) can be written as

$$f_x.u + f_y.v = -f_t \tag{2.9}$$

Lucas and Kanade has assumed that motion is smooth locally i.e. motion vectors in a given region do not change but merely shift from one position to another. For a given pixel we look around its $n \times n$ neighbor with $n > 1$ and assume optical flow on these pixel is same. For example, consider a 3×3 window, the set of equations are,

$$f_{x_1}.u + f_{y_1}.v = -f_{t_1} \tag{2.10}$$

$$f_{x_2}.u + f_{y_2}.v = -f_{t_2} \tag{2.11}$$

$$\vdots$$

$$f_{x_2}.u + f_{y_2}.v = -f_{t_9} \tag{2.12}$$

This system of equations can be written as:

$$\begin{bmatrix} f_{x_1} & f_{y_1} \\ \vdots & \vdots \\ f_{x_9} & f_{y_9} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f_{t_1} \\ \vdots \\ -f_{t_9} \end{bmatrix} \tag{2.13}$$

$$AU = f_+ \tag{2.14}$$

where $A = \begin{bmatrix} f_{x_1} & f_{y_1} \\ \vdots & \vdots \\ f_{x_9} & f_{y_9} \end{bmatrix}$, $U = \begin{bmatrix} u \\ v \end{bmatrix}$, $f_+ = \begin{bmatrix} -f_{t_1} \\ \vdots \\ -f_{t_9} \end{bmatrix}$

Now, vector U can be computed using the pseudo inverse method as follows:

$$A'AU = A'f_+$$

$$U = (A'A)^{-1}A'f_+ \tag{2.15}$$

(c) **Least Square Fit Method (LSF)**: Ideally equation (2.13) should be zero but it is not happening i.e. error are there because we are estimating u and v. In some equation it is positive and in some it is negative so we square the error and sum it.

$$minimize \sum_{i=1}^{n^2} (f_{x_i}u + f_{y_i}v + f_{t_i})^2 \tag{2.16}$$

Differentiating equation (2.16) with respect to u and v separately, final equation we get

$$\sum (f_{x_i}u + f_{y_i}v + f_{t_i})f_{x_i} = 0 \tag{2.17}$$

$$\sum (f_{x_i}u + f_{y_i}v + f_{t_i})f_{y_i} = 0 \tag{2.18}$$

This system of equations can be written as:

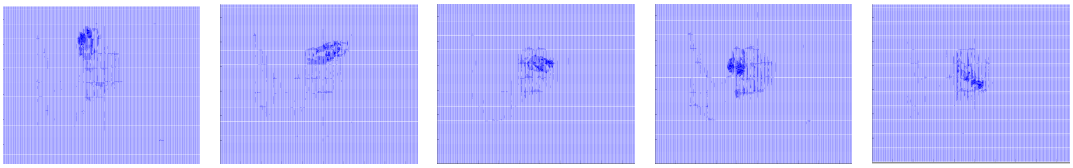
$$\begin{bmatrix} \sum (f_{x_i})^2 & \sum f_{x_i}f_{y_i} \\ \sum f_{x_i}f_{y_i} & \sum (f_{y_i})^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum f_{x_i}f_{t_i} \\ -\sum f_{y_i}f_{t_i} \end{bmatrix} \tag{2.19}$$

$$BU = f \tag{2.20}$$

where, $B = \begin{bmatrix} \sum (f_{x_i})^2 & \sum f_{x_i} f_{y_i} \\ \sum f_{x_i} f_{y_i} & \sum (f_{y_i})^2 \end{bmatrix}$, $U = \begin{bmatrix} u \\ v \end{bmatrix}$, $f = \begin{bmatrix} -\sum f_{x_i} f_{t_i} \\ -\sum f_{y_i} f_{t_i} \end{bmatrix}$

Equation (2.20) gives the optical flow of each pixel.

Horn and Schunck method gives the global information and smooth flow, whereas non-iterative Lucas and Kanade method gives the local information. The latter one does not yield a very high density of flow vectors. Least square fit is an extension of Lucas and Kanade method which minimizes the error produced by LKW method. Due to smooth and higher density of flow vector, the method of Horn and Schunck is found to perform well and hence discussed further in detail. The results, based on all the three methods are although presented at the end of subsection 2.3.2 for comparative analysis. The optical flow estimated by Horn and Schunck method of the preprocessed frames of the video is given in Figure 2.4 and suitable thresholding is performed to get the region of interest.



(a) Frame 1 (b) Frame 20 (c) Frame 30 (d) Frame 60 (e) Frame 83

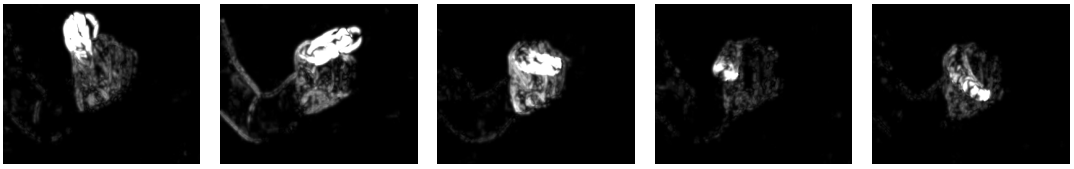
Figure 2.4: Optical flow between frame (1, 2), frame (20, 21), frame (30, 31), frame (60, 61), and frame (83, 84).

2.2.2 Thresholding

For each pixel, u and v are the optical flow in x and y directions respectively. The magnitude of optical flow for each pixel is given by

$$M = \sqrt{u^2 + v^2} \tag{2.21}$$

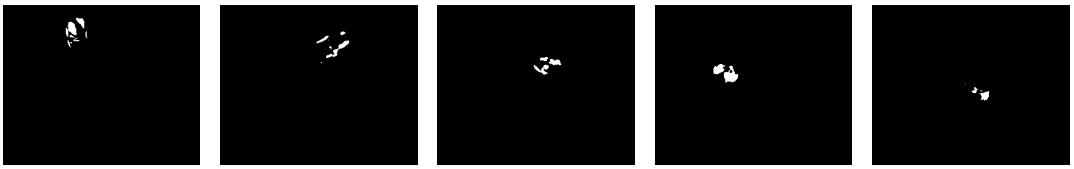
To study the motion of each pixel the magnitude M has been assigned as pixel intensity values, thus resulting in a sequence of gray scale images as shown in Figure 2.5. Value of τ is found using Otsu method [38].



(a) Frame 1 (b) Frame 20 (c) Frame 30 (d) Frame 60 (e) Frame 83

Figure 2.5: Gray scale form of finger gesture.

Higher the value of u and v, the higher is the magnitude M of motion and hence more prominent is the pixel motion in the corresponding gray scale image. Further the region of interest is segmented using Algorithm 2 and shown in Figure 2.6.



(a) Frame 1 (b) Frame 20 (c) Frame 30 (d) Frame 60 (e) Frame 83

Figure 2.6: Frames showing prominent motion after thresholding.

Morphological operation is performed to remove the unwanted motion which is still left after motion segmentation and the result is shown in Figure 2.7. The next step is to convert the image sequence in static shape pattern and it is achieved using motion history image [8].

(a) Frame 1 (b) Frame 20 (c) Frame 30 (d) Frame 60 (e) Frame 83

Figure 2.7: Frames after removal of unwanted motion.

Algorithm 2 Image intensity representation and thresholding

Input : Computed (u, v) as pixel velocity components in x and y direction respectively
Output : Prominent motion after thresholding
for $i \leftarrow 1$ *to height of frame* **do**
  for $j \leftarrow 1$ *to width of frame* **do**
    $z_u(i,j) \leftarrow u(i,j) * u(i,j)$
    $z_v(i,j) \leftarrow v(i,j) * v(i,j)$
    $mag(i,j) \leftarrow \sqrt{(z_u(i,j) + z_v(i,j)}$
  end for
end for
$q1 \leftarrow maximum\ (mag)$
$q2 \leftarrow minimum\ (mag)$
for $i \leftarrow 1$ *to height of frame* **do**
  for $j \leftarrow 1$ *to width of frame* **do**
    $MAG(i,j) \leftarrow\ mag(i,j)/(q1 - q2)$
    if $MAG(i,j)\ <= \tau$ **then**
      $MAG(i,j) \leftarrow 0$
    end if
  end for
end for

2.3 Motion History Image (MHI)

The motion history image (MHI) approach is a view-based temporal template method which is simple but robust in representing movements and is widely employed by various research groups for action recognition, motion analysis and other related applications [8]. It describes *how* the object is moving and records the temporal history of motion. Approaches based on template matching first convert an image sequence into a static shape pattern and then compare it to the pre-stored action prototypes during recognition. In the MHI, the silhouette sequence is condensed into gray scale images, while dominant motion information is preserved. Therefore, it can represent a motion sequence in compact manner. This MHI template is also not so sensitive to silhouette noises, like holes, shadows, and missing parts. It keeps a history of temporal changes at each pixel location, which then decays over time [39]. The MHI expresses the motion flow or sequence by using the intensity of every pixel in temporal manner. One of the advantages of MHI is that a range of times may be encoded in a single frame. It spans the time scale of human gestures. The motion

history recognizes general patterns of movement; thus, it can be implemented with cheap cameras and lower powered CPUs [40]. This method does not need trajectory analysis [41]. The MHI is computed using update function ψ(x,y,t), which represents the precomputed optical flow.

$$MH(x,y,t) = \begin{cases} \tau & if \quad \psi(x,y,t) = 1 \\ \max(0, MH(x,y,t-1) - \delta) & otherwise \end{cases}$$

where, (x, y) and t shows position and time, τ is the temporal extent of the movement for example number of frames, and δ is the decay operator. The result of this computation is a scalar-valued image, where more recently moving pixels are brighter and vice-versa [2, 42]. The recursive definition implies that no history of the previous images or their motion fields needs to be stored nor manipulated, which makes the computation fast and space efficient. Figure 2.8 shows the formation of motion history image of numeral '2'.

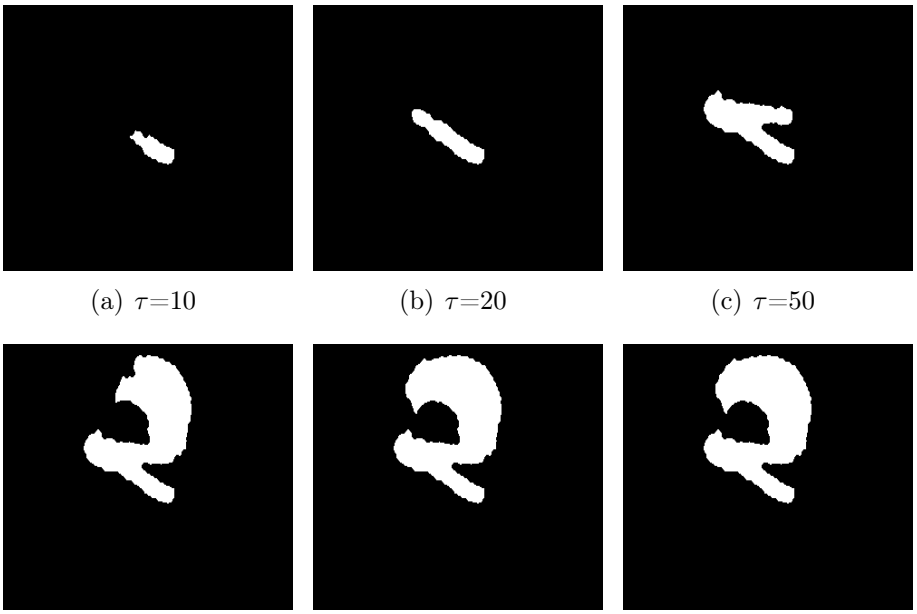
(a) Frame 1 (b) Frame 20 (c) Frame 30 (d) Frame 60 (e) Frame 83

Figure 2.8: Frames showing the formation of MHI.

2.3.1 Effect of τ and δ on MHI

Different MHI is produced at different τ values. If the τ is smaller than the number of frames, then the prior information of the gesture is lost in its MHI. Figure 2.9 shows the dependence on τ in producing the MHI. For example, when τ =20 for a gesture having 83 frames, there is a loss of motion information after 20 frames where the value of decay parameter δ is 1. On the other hand, if the temporal duration value is set at very high value compared to the number of frames, for example 230 in this case then the changes of pixel values in the MHI template is less significant. Figure 2.10 shows the dependence on decay parameter δ while calculating the MHI image. If there is no

change of motion in a specific pixel where earlier there was a motion, the pixel value is reduced by δ. However, having different δ values may provide slightly different information; hence the value can be chosen empirically. It is evident from Figure 2.10 that higher values for δ remove earlier trail of motion sequence. This information is important based on the demand and action, we can modulate the value of δ and τ. In our work the value of τ and δ is taken as 230 and 1 empirically for the formation of MHI.



(a) τ=10 (b) τ=20 (c) τ=50

(d) τ=75 (e) τ=83 (f) τ=230

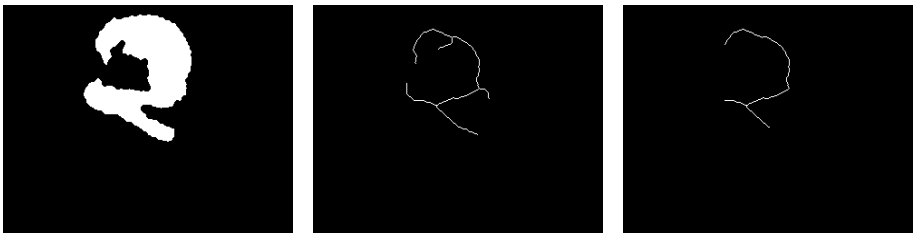
Figure 2.9: Effect of τ in calculating MHI template where δ=1.

(a) δ=1 (b) δ=4 (c) δ=7

Figure 2.10: Effect of δ in calculating MHI template.

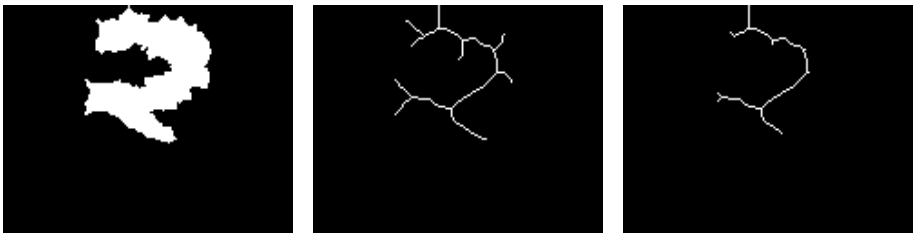
2.3.2 Post processing

Thickness of the gesture may differ among different samples, so to bring uniformity thinning [43, 44] is performed. Thinning is a morphological operation in which binary valued image regions are reduced to lines that approximate the center skeletons of the regions [24]. It outputs the thinnest representation of object that preserve the topology aiding synthesis as shown in Figure 2.11(b). Unwanted spurs [24] are removed by setting the pixel value to black using the pruning operation which is shown in Figure 2.11(c). Similarly, using the above mentioned steps Figure 2.12 and 2.13 show the formation of the numeral '2' using Lucas and Kanade Window method, Least Square Fit method respectively, which clearly shows the formation of MHI using Horn and Schunck method gives better result. The other numeral are formed in which Horn and Schunck optical flow method is used, shown in Figure 2.14.



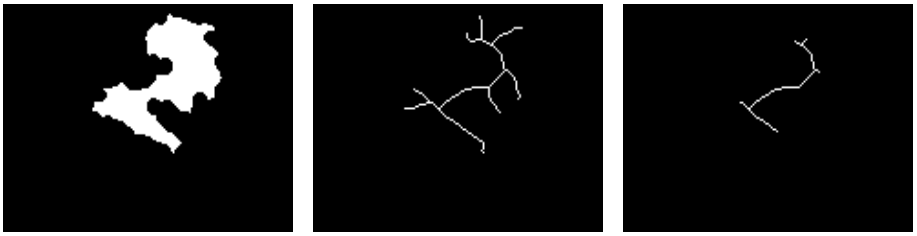
(a) (b) (c)

Figure 2.11: Final post processing on MHI (Horn and Schunck method).



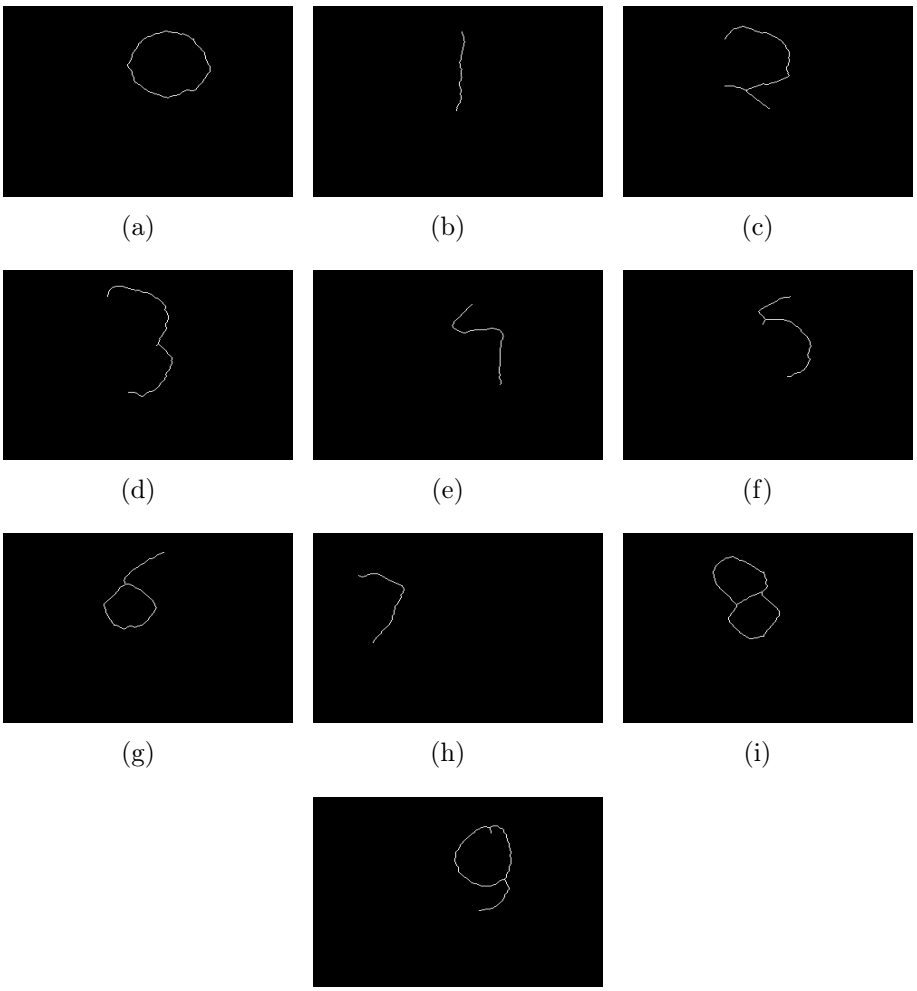
(a) (b) (c)

Figure 2.12: Final post processing on MHI (Lucas and Kanade Window method).



(a) (b) (c)

Figure 2.13: Final post processing on MHI (Least Square Fit method).



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j)

Figure 2.14: From (a) – (j) frames showing English numeral 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 respectively.

2.4 Summary

In this chapter the formation of numeral using finger gesture is presented where acquisition of video is done using mobile camera having resolution 5M Pixel, so acquisition device is not a limitation. In the video motion of index finger is captured which is obtained using the three different optical flow method presented in this chapter. Temporal history of motion is recorded using motion history image and finally post processing is done to get a better thinned image.

Chapter 3

Formation of Numeral using Pointer with a Colored Tip

In this chapter, we have proposed a scheme, in which numeral gesture is formed by some external means like a pen whose tip is either red, green, or blue. Input video is captured using a mobile camera with 5M pixel resolution. Figure 3.1 shows the frames of a video in which gesture of numeral '5' is performed using a pen whose tip is red. The input video is preprocessed to convert it into its true form using Algorithm 1 and the result is shown in Figure 3.2.

(a) Frame 1 (b) Frame 26 (c) Frame 57 (d) Frame 72 (e) Frame 90

Figure 3.1: Frames of original video.

(a) Frame 1 (b) Frame 26 (c) Frame 57 (d) Frame 72 (e) Frame 90

Figure 3.2: Frames of preprocessed video.

In the previous chapter, we have performed segmentation using intensity feature

only. Here our objective is to achieve motion segmentation using both color and brightness information.

3.1 Color Segmentation

Color is perceived as a combination of three color stimuli: red, green, and blue, which forms a color space. RGB colors are called primary colors and are additive. Figure 3.3 shows the RGB color model. By varying their combinations, other colors can be obtained. Color is characterized by three quantities.

- **Hue**: It is an attribute that defines pure color. It is associated with the dominant wavelength in a mixture of light waves. It represents the dominant color perceived by observer, i.e whenever we call an object red, green or blue, we refer to its hue.
- **Saturation**: It gives a measure of degree to which a pure color is diluted with white light. It is inversely proportional to the amount of white light added.
- **Brightness**: It is the achromatic notion of intensity and is one of the key factor to describe color sensation.

A color model is a specification of a co-ordinate system within which each color is represented by a single point. The RGB space does not lend itself to mimic the higher level processes which demand the perception of color with respect to the human visual system. It is better represented in terms of hue, saturation, and intensity [20, 45, 46]. One example of such representation is the HSI color space.

3.1.1 HSI (Hue, Saturation, Intensity) color model

The HSI color space, decouples the intensity component from the color carrying information (hue and saturation) in a color image [20]. An RGB color image is composed of three monochrome intensity images, in which intensity is extracted from RGB image. The color cube shown in Figure 3.3 stands along the black,(0,0,0), vertex, with the white vertex,(1,1,1), directly above it, as shown in Figure 3.4. The intensity is along the line joining these two vertices. To determine the intensity component of

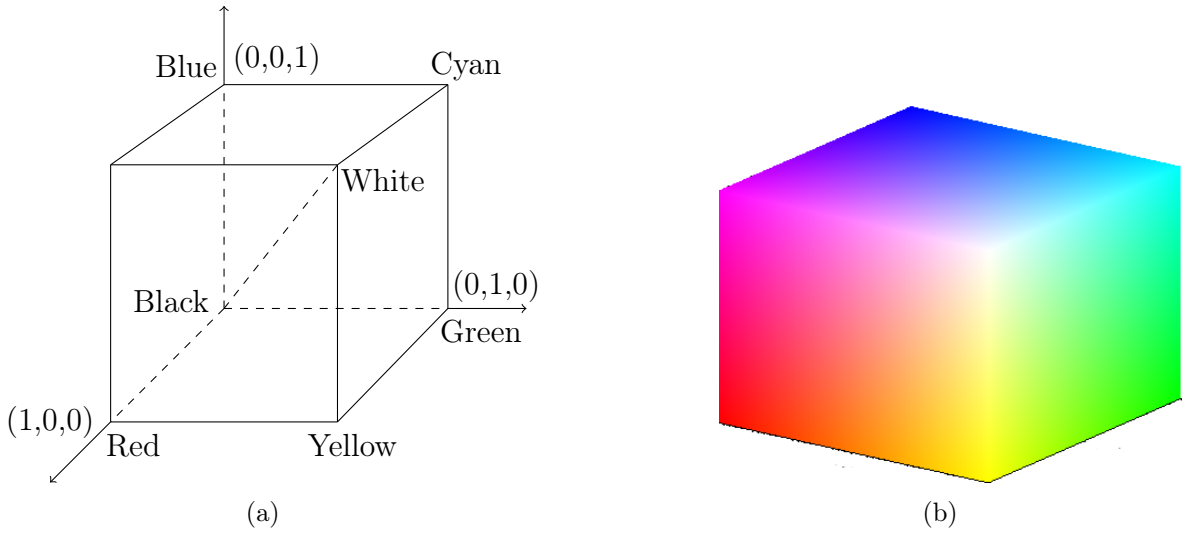


Figure 3.3: (a) Schematic of the RGB color cube showing the primary and secondary colors of light at vertices. Points along the main diagonal have gray value from black at the origin to white at point (1,1,1). (b) The RGB color cube.

any color point, a plane is passed perpendicular to the intensity axis, containing the color point. The intersection of the plane with the intensity axis gives the intensity value. Saturation of a color increases as a function of distance from intensity axis. The saturation of points on the intensity axis is zero, as evidenced by the fact that all points along the axis are shades of gray. In order to understand how hue can be determined from a given RGB point, consider Figure 3.4(b), which shows a plane defined by three points, (black, white, and cyan). The black and white points contained in the plane illustrate that intensity axis is also contained in that plane. All points contained in the plane segment defined by the intensity axis and the boundaries of the cube have same hue. This is because the colors inside a color triangle are various combinations or mixtures of the three vertex colors. If two of those vertices are black and white, and third is a color point, all points on the triangle must have the same hue because black and white components do not contribute to changes in hue. By rotating the shaded plane about the vertical intensity axis, different hue value are obtained.

The HSI space consists of a vertical intensity axis and the locus of color points that lie in a plane perpendicular to this axis. As the plane moves up and down the intensity axis, the boundaries defined by the intersection of the plane with the faces of the cube have either a triangular or hexagonal shape. This can be visualized much

more by looking at the cube down its gray scale axis, as shown in Figure 3.5(a). In this plane primary colors are separated by 120°. The secondary colors are 60° from the primaries, which means angle between secondary colors is 120°.

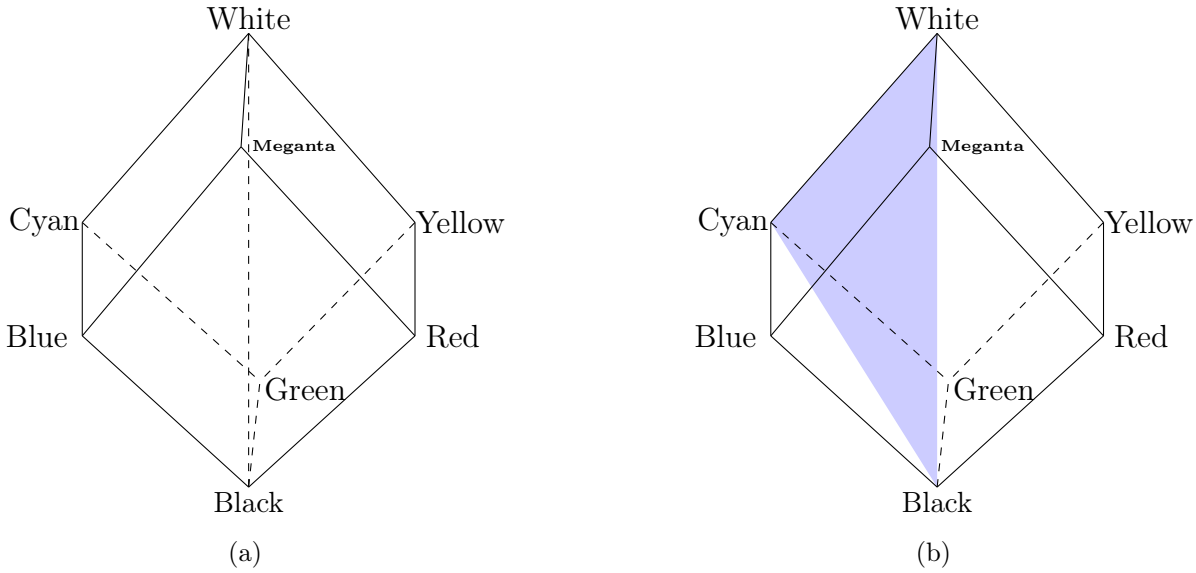


Figure 3.4: Relationship between RGB and HSI color model.

Figure 3.5(b) shows the hexagonal shape and an arbitrary color point (shown as a dot). The hue of the point is determined by an angle from some reference point. Usually an angle of 0° from the red axis designates 0 hue, and increases counterclockwise subsequently. The saturation is the length of vector from the origin to the point. The origin is defined by the intersection of the color plane with the vertical intensity axis. The important components of The HSI color space are the vertical intensity axis, the length of the vector to a color point, and the angle this vector makes with the red axis.

Given an image in RGB color format is converted into HSI model as,

$$H = \begin{cases} \theta & if \quad B \leq G \\ 360 - \theta & if \quad B > G \end{cases} \tag{3.1}$$

where

$$\theta = \cos^{-1}\left\{ \frac{0.5 \times [(R-G)+(R-B)]}{[(R-G)^2 + (R-B)(G-B)]^{1/2}} \right\}$$

$$S = 1 - \frac{3}{(R+G+B)}[minimum(R,G,B)] \tag{3.2}$$

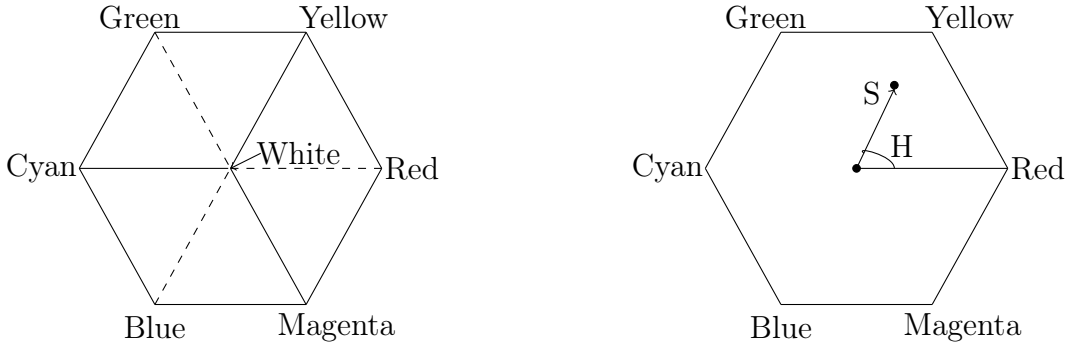


Figure 3.5: Hue and saturation in HSI color model.

$$I = \frac{1}{3}(R + G + B) \tag{3.3}$$

RGB color model is converted into HSI color space using equations (3.1), (3.2), and (3.3) as shown in Figure 3.6. The segmentation of color is carried out using Algorithm 3, and the result is shown in Figure 3.7.

(a) Frame 1 (b) Frame 26 (c) Frame 57 (d) Frame 72 (e) Frame 90

Figure 3.6: Frames of video in HSI color model.

Algorithm 3 Color segmentation

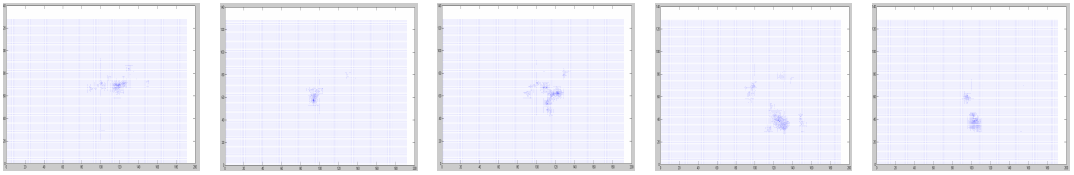
Input : Preprocessed video U and video L in HSI color model
Output : Segmented video W in RGB color model
for $i \leftarrow 1$ *to number of frame* **do**
  for $j \leftarrow 1$ *to height of frame* **do**
    for $k \leftarrow 1$ *to width of frame* **do**
      if $L(j,k,1,i) > \tau_1$ *and* $L(j,k,2,i) > \tau_2$ *and* $L(j,k,3,i) > \tau_3$ **then**
        $W(j,k,:,i) \leftarrow U(j,k,:,i)$
      else
        $W(j,k,:,i) \leftarrow 0$
      end if
    end for
  end for
end for

(a) Frame 1 (b) Frame 26 (c) Frame 57 (d) Frame 72 (e) Frame 90

Figure 3.7: Frames of video after color segmentation.

3.2 Motion Segmentation and Motion History Image (MHI) formation

The motion of the tip is obtained using the Horn and Schunck optical flow method [9] as shown in Figure 3.8. The Motion of the tip is segmented using Algorithm 2 whose result is shown in Figure 3.9. Some unwanted motion is removed using morphological operations as shown in Figure 3.10.



(a) Frame 1 (b) Frame 26 (c) Frame 57 (d) Frame 72 (e) Frame 89

Figure 3.8: Optical flow between frame (1, 2), frame (26, 27), frame (57, 58), frame (72, 73), and frame (89, 90).

(a) Frame 1 (b) Frame 26 (c) Frame 57 (d) Frame 72 (e) Frame 89

Figure 3.9: Frames showing prominent motion after thresholding.

As we know the MHI is used to record the temporal history of motion. Here the MHI is formed using any three primary color red, green, or blue as shown in

(a) Frame 1 (b) Frame 26 (c) Frame 57 (d) Frame 72 (e) Frame 89

Figure 3.10: Binary image after removal of unwanted motion.

Figure 3.11, 3.13, and 3.15. The value of τ and δ is taken as 230 and 1 empirically.

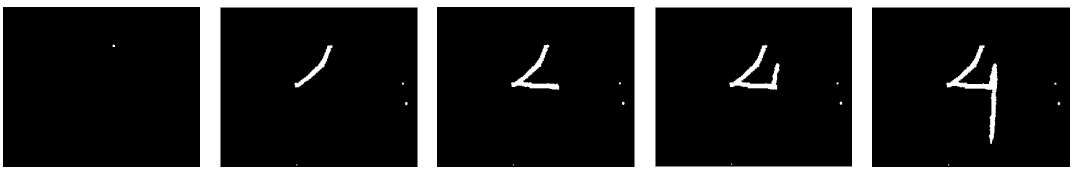
(a) Frame 1 (b) Frame 26 (c) Frame 57 (d) Frame 72 (e) Frame 89

Figure 3.11: Frames of MHI using red color.

- **MHI using green color**

(a) Frame 1 (b) Frame 28 (c) Frame 51 (d) Frame 69 (e) Frame 93

Figure 3.12: Frames of preprocessed video using green color.



(a) Frame 1 (b) Frame 28 (c) Frame 51 (d) Frame 69 (e) Frame 92

Figure 3.13: Frames of MHI using green color.

- **MHI using blue color**

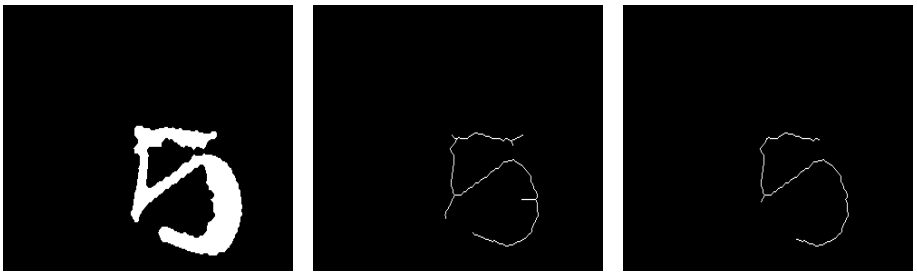
(a) Frame 1 (b) Frame 29 (c) Frame 51 (d) Frame 69 (e) Frame 99

Figure 3.14: Frames of preprocessed video using blue color.

(a) Frame 1 (b) Frame 29 (c) Frame 51 (d) Frame 69 (e) Frame 98

Figure 3.15: Frames of MHI using blue color.

Thinning [24] is performed to bring uniformity as the thickness of the gesture may differ among different samples. To get better thinned image, the holes are filled shown in Figure 3.16(a). Unwanted spurs [24] are removed by setting the pixel value to black using the pruning operation and the result is shown in Figure 3.16(c).



(a) (b) (c)

Figure 3.16: Final post processing on MHI.

Likewise, the other English numeral is formed shown in Figure 3.17. Using this scheme numeral of another language can be formed as Figure 3.18 shows the Odia numeral.

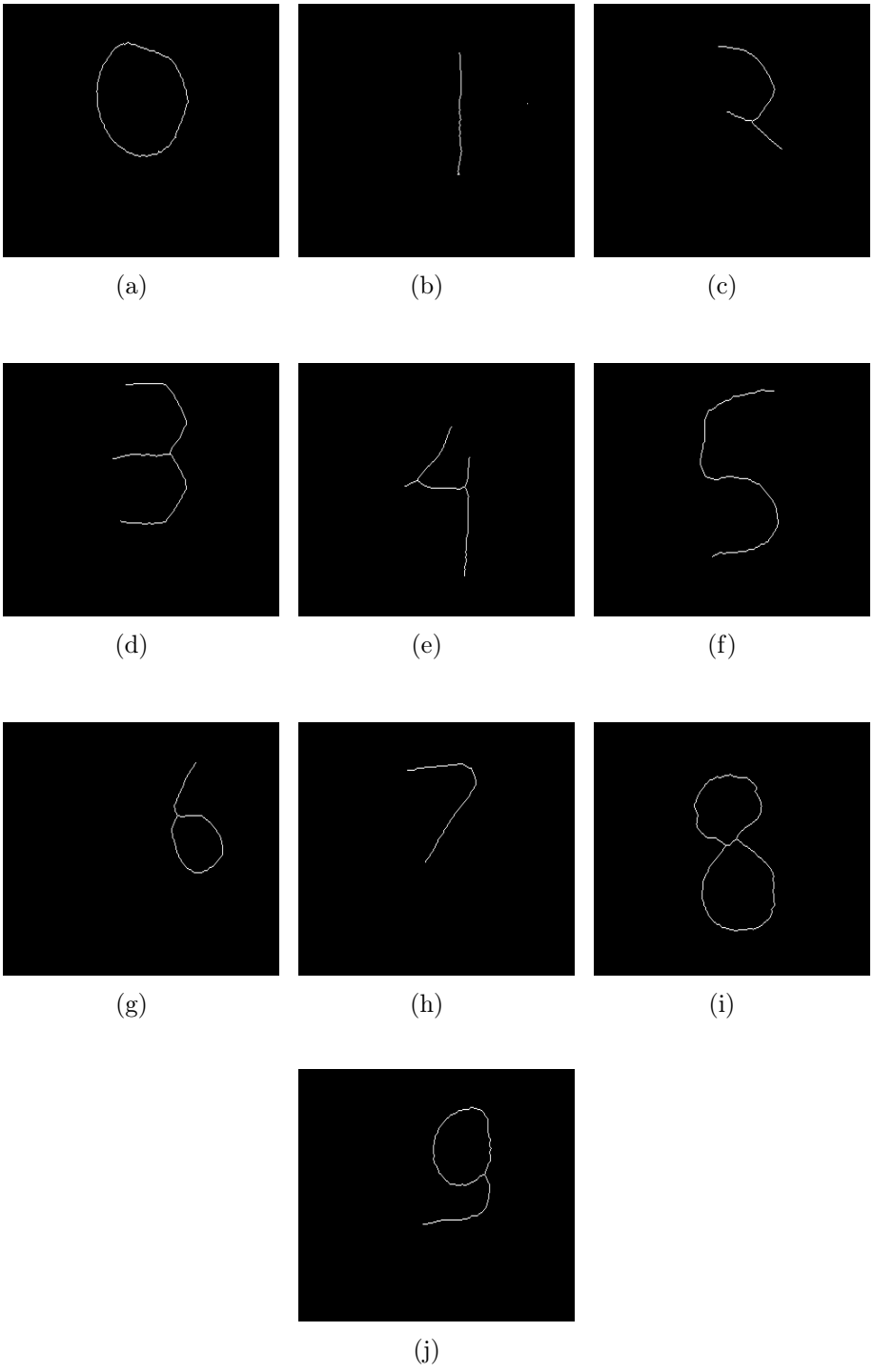


Figure 3.17: From (a)–(j) frames showing English numeral 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 respectively.

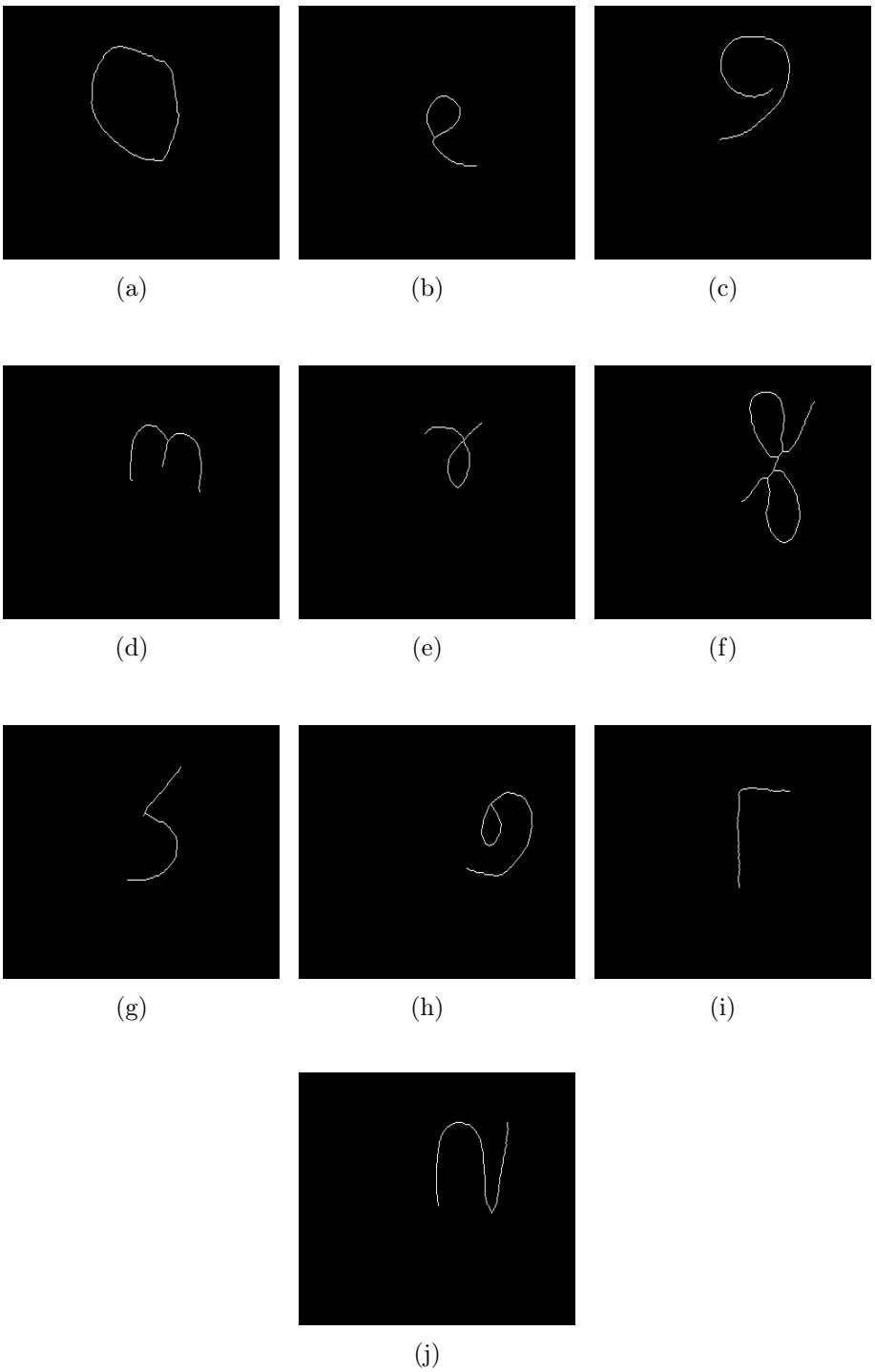


Figure 3.18: From (a)–(j) frames showing Odia numeral 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 respectively.

3.3 Summary

In this chapter, we have suggested a scheme for the formation of numeral using both color and brightness as features. The proposed scheme is invariant to other unwanted

motion that may arise in the surrounding. Gesture is performed with the help of external means like a pen whose tip is either red, green, or blue. In particular, any finger can also be used with a colored tip in it. HSI color model is used to segment the colored tip followed by optical flow to segment the motion and finally motion history image is obtained. Using this approach numeral of another language can also be formed.

Chapter 4

Feature Extraction and Recognition

Features are the inherent property of data. Transforming the input data into the set of features is called feature extraction. Feature extraction involves simplifying the amount of information required to describe input data.

4.1 Feature Extraction

In this thesis, we have used geometrical property for feature vector representation of each numeral. These geometrical properties are extracted from binary image of the segmented numeral at different depth of spatial resolution through a hierarchical abstraction of the image data [47, 48]. All images are scaled to a fixed size of $w \times w$ before feature extraction. For simulation purpose images are scaled to 128×128. The scaled image is then divided into sub-images, based on the k-d tree splitting strategy, which is a multi-dimensional binary search tree [47, 49]. It is a recursive partitioning tree in which the partition is done along the x and y axis in alternative fashion. Each such partition of an image into two sub-images, define the depth of k-d tree decomposition. At each depth p total number of sub-image is 2^p as illustrated in Figure 4.1. The decomposition at $p = 1$ is done by dividing the image into two sub-images along the y-axis. Decomposition at subsequent depth are done by calling itself recursively on the transpose of each sub-images. Similarly splitting of image into sub-images at different depth is done along x-axis. Centroid of each sub-image is calculated relative to the centroid of the complete image and these values are normalized, dividing it by the number of rows or columns. These values are taken

as a feature vector. At each depth p, the number of feature points is $2^{p+1} - 4$. The dimension of feature vector depends on the value of p.

(a) $p = 1$ (b) $p = 2$ (c) $p = 3$

Figure 4.1: Image division based on K-d tree decomposition.

In the Figure 4.2, the numeral image is divided along y-axis at $p = 1$, gives two sub-images. At $p = 2$ the sub-images are divided along x-axis according to k-d tree decomposition. The result of such splitting gives the two feature point $y_0 - y_1$ and $y_0 - y_2$. Similarly, another two feature points $x_0 - x_1$ and $x_0 - x_2$ are obtained when the division of numeral image is done along x-axis at $p = 1$. Feature vector is normalized by dividing it by the number of rows or columns.

$$Feature\ vector = \left[\frac{y_0 - y_1}{w}, \frac{y_0 - y_2}{w}, \frac{x_0 - x_1}{w}, \frac{x_0 - x_2}{w}\right]$$

where w is the number of rows or columns.

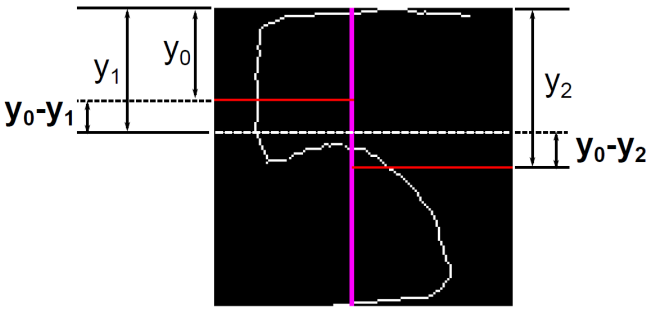


Figure 4.2: Illustration of feature vector for $p = 2$.

4.2 Recognition

In this section, we have discussed an approach for numeral recognition, which is achieved by finding the minimum distance between the query image and each stored template. Any distance measure for the purpose of object matching should have the following properties: (a) It should have a large discriminatory power (b) its value should increase with the amount of difference between the two objects. One such distance measure is Modified Hausdorff Distance (MHD) [50]. Figure 4.3 shows the block diagram for the recognition of numeral. The MHD is calculated between feature vector f_Q of query image and f_0 , f_1 , ... f_9 of stored templates. It gives ten distances d_0 , d_1 , ... f_9 , where each subscript represents the respective numeral. The subscript of the minimum distance gives the recognized numeral.

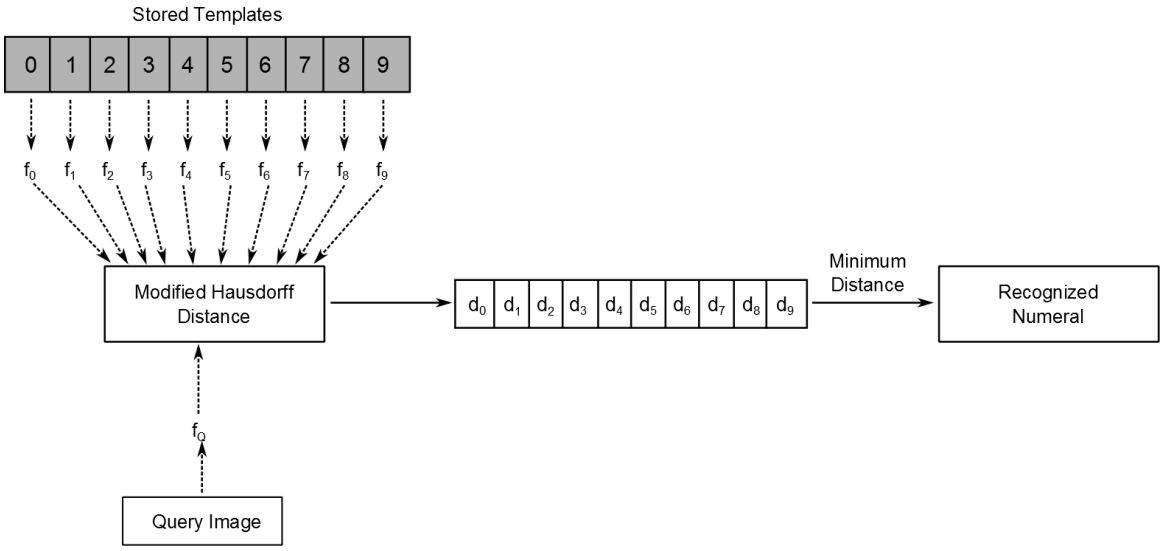


Figure 4.3: Block diagram for the recognition of numeral.

For any given two set of points $A = (a_1, ..., a_{N_a})$ and $B = (b_1, ..., b_{N_b})$, MHD is given by,

$$\max(d(A, B), d(B, A))$$

where

$$d(A, B) = \frac{1}{N_a} \sum_{a \in A} d(a, B)$$

$$d(B, A) = \frac{1}{N_b} \sum_{b \in B} d(b, A)$$

$$d(a, B) = min_{b \in B}||a - b||$$

$$d(b, A) = min_{a \in A}||b - a||$$

N_a , N_b is number of element in A and B respectively. $\|a - b\|$ is the Euclidean distance between two points a and b.

4.2.1 Accuracy matrix

For measuring accuracy we adopted different metrics, namely *Sensitivity, Specificity, Precision,* F_1 *score, Percentage of Correct Classification (PCC).*

Sensitivity, also called the true positive rate or the recall rate, measures the percentage of true positives which are correctly identified, and is complementary to the false negative rate. Sensitivity relates to the test's ability to identify a condition correctly.

$$Sensitivity = \frac{TP}{TP + FN} \tag{4.1}$$

Specificity, sometimes called the true negative rate, measures the percentage of true negatives which are correctly identified, and is complementary to the false positive rate. Specificity relates to the test's ability to exclude a condition correctly.

$$Specificity = \frac{TN}{FP + TN} \tag{4.2}$$

Precision, also known as positive predictive value, measures the percentage of the true positives against all the positive results (both true positives and false positives).

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

F_1 **score**, also known as Figure of Merit or F-measure, that is the weighted harmonic mean of Precision and Recall.

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4.4}$$

PCC, Percentage of correct classification, is used as the measure for accuracy, and is defined as

$$PCC = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.5}$$

where TP is true positive that represents the number of correctly matched input, and TN is true negative representing the number of correctly rejected input. Similarly FP is false positive that represents the number of incorrectly matched input, and FN is false negative representing the number of incorrectly rejected inputs.

4.2.2 Accuracy Results

To evaluate the performance of our suggested scheme, 20 samples each of English and Odia numeral are taken. The above performance measures are computed for both English and Odia numerals at various depths (depth = 2,3,4,5,6). We found experimentally that the maximum accuracy is obtained at depth 5 and 4 for English and Odia numeral respectively, as given in Table 4.1, and Table 4.2 respectively.

Table 4.1: Accuracy metric for English numeral at depth = 5

Numeral	**Sensitivity**	**Specificity**	**Precision**	**F1 Score**	**PCC**
0	100	100	100	100	100
1	60	100	100	75	96
2	100	100	100	100	100
3	70	90	43.75	53.85	88
4	70	97.78	77.78	73.68	95
5	80	97.78	80	80	96
6	60	95.56	60	60	92
7	100	100	100	100	100
8	60	93.33	50	54.55	90
9	80	97.78	80	80	96

Table 4.2: Accuracy metric for Odia numeral at depth = 4

Numeral	**Sensitivity**	**Specificity**	**Precision**	**F1 Score**	**PCC**
0	100	100	100	100	100
1	100	100	100	100	100
2	90	98.89	90	90	98
3	90	96.67	75	81.82	96
4	100	98.89	90.91	95.24	99
5	100	100	100	100	100
6	90	98.89	90	90	98
7	90	98.89	90	90	98
8	100	100	100	100	100
9	70	100	100	82.35	97

4.3 Summary

In this chapter centroid of sub-images are calculated relative to the centroid of complete image which is taken as a feature vector. Decomposition of image into sub-images are done using k-d tree splitting method and MHD is used for recognition.

Chapter 5

Conclusion

In this thesis, we propose feature extraction and recognition of numerals specified through gesture. Motion of the index finger is captured using a mobile camera having resolution 5M pixel and its motion is identified using optical flow method. Three different optical flow methods namely, Horn and Schunck, Lucas and Kanade, and Least Square Fit method are used, in which Horn and Schunck optical flow method has been shown to give the better results. Motion History Image (MHI) template are generated to get the numeral. Different gesture have different thickness, so to bring uniformity thinning is performed and the unwanted parasitic components are removed. Further gestures are formed using a pen whose tip is either red, green, or blue. In the scene multiple persons are present performing other activities which is not affecting the final result of recognition. The video captured is in RGB color model which is converted into HSI color model and the motion of the tip of pen is segmented using Horn and Schunck optical flow method. After getting the preprocessed numeral image, it is divided into sub-images using k-d tree decomposition method. Centroid of the sub-images is calculated relative to the centroid of complete image which is taken as feature vector and Modified Hausdorff Distance (MHD) is used for recognition.

Above mentioned approach can be applied to the recognition of character in any language. The indoor environment under the uniform illumination can be extended to the outdoor environment with non-uniform illumination.

Bibliography

[1] J. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 16, 2011.

[2] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257 – 267, 2001.

[3] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 3, pp. 311 – 324, 2007.

[4] C. Shan, Y. Wei, X. Qiu, and T. Tan, "Gesture recognition using temporal template based trajectories," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR), 2004.*, vol. 3. IEEE, 2004, pp. 954 – 957.

[5] M. Ishikawa and H. Matsumura, "Recognition of a hand-gesture based on self-organization using a data glove," in *6th International Conference on Neural Information Processing, ICONIP'99.*, vol. 2. IEEE, 1999, pp. 739 – 745.

[6] M. A. Qureshi, A. Aziz, M. A. Saeed, M. Hayat, and J. S. Rasool, "Implementation of an efficient algorithm for human hand gesture identification," in *Electronics, Communications and Photonics Conference (SIECPC), 2011 Saudi International.* IEEE, 2011, pp. 1 – 5.

[7] M. K. Sohn, S. H. Lee, D. J. Kim, B. Kim, and H. Kim, "3D hand gesture recognition from one example," in *IEEE International Conference on Consumer Electronics (ICCE) 2013.* IEEE, 2013, pp. 171 – 172.

[8] M. Ahad, A. Rahman, J. Tan, H. Kim, and S. Ishikawa, "Temporal motion recognition and segmentation approach," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 91 – 99, 2009.

[9] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1, pp. 185 – 203, 1981.

[10] D. Zhang and G. Lu, "Segmentation of moving objects in image sequence: A review," *Circuits, Systems and Signal Processing*, vol. 20, no. 2, pp. 143 – 183, 2001.

[11] P. Hong, M. Turk, and T. S. Huang, "Gesture modeling and recognition using finite state machines," in *4th IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, 2000, pp. 410 – 415.

[12] J. Lei, Z. Cheng, and W. Junbo, "A recognition method for one-stroke finger gestures using a mems 3D accelerometer," *IEICE transactions on information and systems*, vol. 94, no. 5, pp. 1062 – 1072, 2011.

[13] C. Oz and M. C. Leu, "American sign language word recognition with a sensory glove using artificial neural networks," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 7, pp. 1204 – 1213, 2011.

[14] K. S. Fu and J. Mui, "A survey on image segmentation," *Pattern recognition*, vol. 13, no. 1, pp. 3 – 16, 1981.

[15] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern recognition*, vol. 26, no. 9, pp. 1277 – 1294, 1993.

[16] R. Jain, R. Kasturi, and B. G. Schunck, *Machine vision*. McGraw-Hill New York, 1995, vol. 5.

[17] R. M. Haralick and L. G. Shapiro, "Image segmentation techniques," *Computer vision, graphics, and image processing*, vol. 29, no. 1, pp. 100 – 132, 1985.

[18] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888 – 905, 2000.

[19] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.

[20] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital image processing using MATLAB*. Gatesmark Publishing Knoxville, 2009, vol. 2.

[21] C. Stiller, J. Konrad, and R. Bosch, "Estimating motion in image sequences - a tutorial on modeling and computation of 2D motion," *IEEE Signal Processing Magazine*, vol. 16, 1999.

[22] G. Wyszecki and W. S. Stiles, *Color science*. Wiley New York, 1982, vol. 8.

[23] R. W. G. Hunt, M. R. Pointer, and M. Pointer, *Measuring colour*. John Wiley & Sons, 2011.

[24] C. Di Ruberto, "Recognition of shapes by attributed skeletal graphs," *Pattern Recognition*, vol. 37, no. 1, pp. 21 – 31, 2004.

[25] P. Soille, *Morphological image analysis: principles and applications*. Springer-Verlag New York, Inc., 2003.

[26] R. S. Choras, "Image feature extraction techniques and their applications for CBIR and biometrics systems," *International Journal of Biology and Biomedical Engineering*, vol. 1, no. 1, pp. 6 – 16, 2007.

[27] E. Saber and A. M. Tekalp, "Integration of color, edge, shape, and texture features for automatic region-based image annotation and retrieval," *Journal of Electronic Imaging*, vol. 7, no. 3, pp. 684 – 700, 1998.

[28] L. Zappella, X. Lladó, and J. Salvi, "Motion segmentation: A review," in *Proceedings of the 2008 conference on Artificial Intelligence Research and Development: Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence.* IOS Press, 2008, pp. 398 – 407.

[29] S. S. Beauchemin and J. L. Barron, "The computation of optical flow," *ACM Computing Surveys (CSUR)*, vol. 27, no. 3, pp. 433 – 466, 1995.

[30] A. D. Bimbo and P. Nesi, "Real-time optical flow estimation," in *International Conference on Systems, Man and Cybernetics.* IEEE, 1993, pp. 13 – 19.

[31] B. McCane, K. Novins, D. Crannitch, and B. Galvin, "On benchmarking optical flow," *Computer Vision and Image Understanding*, vol. 84, no. 1, pp. 126 – 143, 2001.

[32] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert, "Highly accurate optic flow computation with theoretically justified warping," *International Journal of Computer Vision*, vol. 67, no. 2, pp. 141 – 158, 2006.

[33] L. Wixson, "Detecting salient motion by accumulating directionally-consistent flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 774 – 780, 2000.

[34] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International journal of computer vision*, vol. 12, no. 1, pp. 43 – 77, 1994.

[35] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in *9th IEEE International Conference on Computer Vision.* IEEE, 2003, pp. 726 – 733.

[36] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision." in *IJCAI*, vol. 81, 1981, pp. 674 – 679.

[37] B. D. Lucas, "Generalized image matching by the method of differences," Ph.D. dissertation, Carnegie Mellon University, 1985.

[38] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[39] T. Xiang and S. Gong, "Beyond tracking: Modelling activity and understanding behaviour," *International Journal of Computer Vision*, vol. 67, no. 1, pp. 21 – 51, 2006.

[40] G. R. Bradski and J. W. Davis, "Motion segmentation and pose recognition with motion history gradients," *Machine Vision and Applications*, vol. 13, no. 3, pp. 174 – 184, 2002.

[41] J. W. Davis, "Sequential reliable-inference for rapid detection of human actions," in *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04).* IEEE, 2004, pp. 111 – 111.

[42] O. Masoud and N. Papanikolopoulos, "A method for human action recognition," *Image and Vision Computing*, vol. 21, no. 8, pp. 729 – 743, 2003.

[43] K. Kaur and M. Sharma, "A method for binary image thinning using gradient and watershed algorithm," *International Journal*, vol. 3, no. 1, 2013.

[44] T. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236 – 239, 1984.

[45] L. Luccheseyz and S. Mitray, "Color image segmentation: A state-of-the-art survey," *Proceedings of the Indian National Science Academy (INSA-A). Delhi*, vol. 67, no. 2, pp. 207 – 221, 2001.

[46] A. K. Jain, *Fundamentals of digital image processing.* Prentice-Hall Englewood Cliffs, 1989, vol. 3.

[47] A. Sexton, A. Todman, and K. Woodward, "Font recognition using shape-based quad-tree and kd-tree decomposition," in *3rd International Conference on Computer Vision, Pattern Recognition and Image Processing*, 2000, pp. 212 – 215.

[48] G. Vamvakas, B. Gatos, and S. J. Perantonis, "Handwritten character recognition through two-stage foreground sub-sampling," *Pattern Recognition*, vol. 43, no. 8, pp. 2807 – 2816, 2010.

[49] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509 – 517, 1975.

[50] M. P. Dubuisson and A. K. Jain, "A modified hausdorff distance for object matching," in *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, vol. 1. IEEE, 1994, pp. 566 – 568.

Dissemination

- **Shree Prakash**, Banshidhar Majhi, Pankaj Kumar Sa "Numeral Recognition from Gesture using K-d Tree Decomposition Scheme", *IEEE International Conference on Electrical, Computer and Communication Technologies(ICECCT 2015)*, Coimbatore [Communicated].

Shree Prakash

Computer Science and Engineering Department,
National Institute of Technology Rourkela,
Rourkela – 769 008, India.

+91 8018260039.

pochaparam@gmail.com

Qualification

- M.Tech. (Research) (CSE) (Continuing)
  National Institute of Technology Rourkela.
- B.Tech. (CSE)
  Bengal Institute of Technology and Management, Santiniketan, [70.02%]
- 12th
  Bihar Intermediate Education Council, Patna, [59%]
- 10th
  Bihar School Examination Board,Patna, [74.8%]

Permanent Address

Sri Thakur Niwas
G. N. Ganj
Po – Laheriasaria
Dist – Darbhanga
State – Bihar Pin – 846001 (India)

Date of Birth

February 25, 1987